2

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

②

Animated Demonstrations
versus
Written Instructions
for Learning Procedural Tasks

Susan Palmiter
Jay Elkerton
Patricia Baggett

Center for Ergonomics
Department of Industrial and Operations Engineering
The University of Michigan

Technical Report C4E-ONR-2
January 1989

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT:

4. PERFORMING ORGANIZATION REPORT NUMBER(S): Tech. Rep. C4E-ONR-2

5. MONITORING ORGANIZATION REPORT NUMBER(S): Tech. Rep. C4E-ONR-2

6a. NAME OF PERFORMING ORGANIZATION: The University of Michigan

6b. OFFICE SYMBOL (If applicable):

7a. NAME OF MONITORING ORGANIZATION: Office of Naval Research

6c. ADDRESS (City, State, and ZIP Code):
Center for Ergonomics
1205 Beal Ave. - IOE Bldg.
Ann Arbor, MI 48109-2117

7b. ADDRESS (City, State, and ZIP Code):
800 N. Quincy Street
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Office of Naval Research

8b. OFFICE SYMBOL (If applicable):

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: Contract: N00014-87-K-0740

8c. ADDRESS (City, State, and ZIP Code):
800 N. Quincy Strret
Arlington, VA 22217-5000

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61153N 42	RR 04209	RR0420901	4429008

11. TITLE (Include Security Classification): (U) Animated Demonstrations versus Written Instructions for Learning Procedural Tasks

12. PERSONAL AUTHOR(S): Susan Palmiter, Jay Elkerton, & Pat Baggett

13a. TYPE OF REPORT: Technical

13b. TIME COVERED: FROM 88-08-15 TO 89-08-14

14. DATE OF REPORT (Year, Month, Day): 89-01-13

15. PAGE COUNT: 44

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number):

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

(U) Direct manipulation interfaces and the need for faster learning have led the development of animated demonstrations so that users can learn interface procedures by watching. To compare animated demonstrations to written instructions we observed users learning and performing HyperCard tasks on an Apple Macintosh computer. Results showed that demonstrations provided faster and more accurate initial learning. However, when the instructions were removed, users of demonstrations took the same and sometimes more time to perform the tasks than did users of written instructions. This finding indicates that demonstrations alone may not provide the necessary knowledge to acquire interface procedures after intial instruction.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: John J. O'Hare

22b. TELEPHONE (Include Area Code): (202) 696-4502

22c. OFFICE SYMBOL: Code 1142PS

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Direct manipulation interfaces and the need for faster learning have led to the development of animated demonstrations so that users can learn interface procedures by watching. To compare animated demonstrations to written instructions we observed users learning and performing HyperCard™ tasks on an Apple® Macintosh™ computer. Results showed that demonstrations provided faster and more accurate initial learning. However, when the instructions were removed, users of demonstrations took the same and sometimes more time tc perform the tasks than did users of written instructions. This finding indicates that demonstrations alone may not provide the necessary knowledge to acquire interface procedures after initial instruction.

INTRODUCTION

When learning to use a computer application, many users are confronted with instructions in a written format. Yet, these instructions are too often discarded because the user wants to get started immediately or because the instructions are difficult to follow and to assimilate (Mack, Lewis, and Carroll, 1983). Often, the user will try to find an expert or colleague who can demonstrate the appropriate interface procedures. Since users may generalize these demonstrated procedures to other tasks, the demonstrations may result in "one-trial" learning (Lewis, Casner, Schoenberg, and Blake, 1987). The purpose of this research is to explore these issues and determine whether demonstrated, animated instructions result in faster learning, better retention, and transfer to similar tasks than written instructions.

Recent interest in using demonstrations has been facilitated by graphical animation software for programming by example (Duisberg, 1988; Myers, 1987). However, "watch me do it" demonstrations have appeared in experimental interfaces such as NLS-SCHOLAR (Grignetti, Hausmann, and Gould, 1975) and CADHELP (Cullingford, Krueger, Selfridge, and Bienkowski, 1982; Neiman, 1982). The NLS-SCHOLAR system used artificial intelligence techniques to teach text editing skills by showing the user how to perform editing tasks. In

CADHELP, an animation program was able to simulate the sequential performance of low-level interface actions by running scripts of these procedures. For example, users could see a demonstration of how the cursor is moved to drag a device. The focus of these efforts has been on the knowledge representations and algorithms required for animation of interface procedures rather than examining the value of demonstrations as a learning aid for users.

Still, demonstrations are being added to software packages more and more often, indicating that interface designers think they are effective ways of teaching interface methods. The Apple Lisa™ and Macintosh™, for example, have introduced "getting started" tours for users. In these tours, basic procedural skills (e.g., pointing and dragging) and more advanced skills (e.g., choosing commands and cutting and pasting) are taught through animated demonstrations, textual instruction, and guided user practice (Apple Computer, 1986). Macintosh application developers have also begun to follow this pattern of using guided tours as a teaching aid (VideoWorks II: MacroMind, 1987). In the VideoWorks guided tour, the user is shown the steps needed to create animated movies.

Unfortunately, little empirical data exist on the efficacy of animated demonstrations for training or helping users. A limited evaluation (6 experienced users) of LisaGuide™ by Carroll and Mazur (1986) suggested that demonstrations may introduce their own usability problems. Therefore, from an engineering perspective, the large amount of work required to develop graphical animations demands that we understand the potential benefits and limitations in usability.

In the current experiment, animated demonstrations were compared with written step-by-step instructions for learning procedural interface tasks. We hypothesized that the two media may be encoded differently as a result of their individual instructional characteristics. One framework hypothesizes that different media are encoded due to different qualities of the instructions as shown in Table 1. Users receiving animated demonstrated lack the explicit,

verbal element of the instruction. Instead, they have two visual codes, one from watching the instructions and one from watching themselves perform those actions. With animated demonstrations users also gain the motoric component of the instruction when actually performing a task. Alternatively, the user who reads instructions and then attempts to perform the task, gets verbal information while reading, and visual and motoric components when performing. This scheme is loosely based on the dual-coding theory set forth by Paivio (1971). We have added the motoric code because evidence suggests that, for procedural tasks, actual manipulation is an important element in learning (see Baggett, 1987).

Table 1

Types of Codes from Animated Demonstrations and Written Instructions When Users See Instructions and Then Perform Them.†

	Verbal	Visual	Motoric
Animated Demonstration		I,U	U
Written Instructions	I	U	U

† An "I" signifies that the instruction itself contains this code. A "U" signifies that the user forms this code when performing a task.

In contrast to written instructions, animated demonstrations may convey procedural knowledge more directly about the interface. Watching an animated demonstration will show the user how the interface appears as the procedure is executed and also links the input actions with the interface results. Users should be able to experience in concrete visual terms how each

procedural step contributes to the overall task goal (Lewis, et al., 1987). Animated demonstrations may allow users to visually rehearse and plan while watching instructions thereby reducing the additional load of forming a motoric code. Animated demonstrations, therefore, may improve initial learning when compared to written directions since the amount of cognitive processing will be reduced during the learning stage.

Reduced processing during initial learning with animated demonstrations is also predicted because demonstrations are integrated into the interface to be learned and serve as examples. As articulated by Anderson, Boyle, Farrell, and Reisner (1984), learning in the problem context should decrease the difficulty of encoding procedural knowledge. A very similar prediction can be made if animated demonstrations are considered as examples of interface procedures. Well constructed examples have been found to improve initial learning in several problem domains (Lewis and Anderson, 1985; Sweller and Cooper, 1985). LeFevre and Dixon (1986) also have found that people rely consistently on examples. LeFevre and Dixon found this preference for examples in a wide variety of conditions and state that people processed written instructions only superficially when examples were available. The reason for users' reliance on examples and performance improvements seems to be the close match between the features of the problem and the example (Lewis, et al., 1985) which may hold true for animated demonstrations. In total, users may be more inclined to use the animated demonstrations and also may find them easier to use.

Still, there is the danger that users may passively watch the animated demonstrations and then blindly mimic these procedures (see LeFevre and Dixon, 1986) with very little processing and encoding. Suppose a user learns a task through an animated demonstration and then is faced with a similar task. Will the animated demonstrations studied earlier aid the user in performing the new task? Some evidence for transfer of training with demonstrated procedures has been provided by Lewis, et al. (1987). However, these studies focused on relatively simple tasks and have not compared demonstrations to other instructional media.

To better understand the demonstration media, we studied animated demonstrations and written directions for moderately complex interface procedures using HyperCard, a hypertext system for the Apple Macintosh. This application was chosen since the object-oriented procedures of HyperCard provide a concrete, direct manipulation interface to fully explore the possible benefits of animated demonstrations. To assess animated demonstrations in comparison to written instructions in different learning phases, we observed users learning and practicing with the instructional media, performing similar and different interface tasks immediately after instruction, and performing these same tasks after a several day delay. In addition to the time and error data collected, users subjective reactions to the instructional media were also assessed.

METHOD

Participants

Twenty-eight students and staff (14 of each gender) from the University of Michigan served as participants. Each was required to have four months experience in using both a graphics and a word processing application, but no experience in using HyperCard on the Macintosh. Table 2 shows the relative experience levels for all participants in this study, illustrating that they had a fair amount of experience using computers.

Experimental Design

The study was a four variable (2 X 2 X 7 X 3), mixed-factor design. The between-subjects variables were the instructional media presented (animated demonstrations or written instructions) and the amount of required practice during instruction (1 or 3 mandatory trials). The practice variable was included to ascertain whether additional performance affected learning and retention of the procedures presented by the two instructional media. The

Table 2

Macintosh and General Computer Experience of Participants.

Experience Measure	Months of use	Frequency per week
Macintosh Experience		
- MacWrite™	12-24 †	1 - 3
- Microsoft® Word	4-8	1 - 3
- MacPaint™	9-12	<1
- MacDraw™	9-12	1 - 3

Experience Measure	Mean Score
Computer Classes - number of classes where computer applications learned	3 - 5
Programming Languages and non-Macintosh Machine Experience	
- number of machines	2.8
- number of languages	2.7

Subjective Rating	Mean Score
Programming Experience (subjective rating assigned by the participant; 1 = beginner, 5 = expert)	2.4
Experience with Programming Languages and non-Macintosh Machines (subjective rating assigned by the participant; 1 = unfamiliar, 5 = expert)	
- Basic	2.6
- Fortran	2.8
- Pascal	2.0
- IBM PC	2.9

† Entries displayed within a range were collected using a forced choice scale.

within-subjects variables were the seven procedural tasks and the three performance assessment sessions in which those tasks were presented. The performance assessment sessions were: 1) a *training session* to determine how participants learned the tasks with the animated or written instructions, 2) an *immediate test session* to test the procedural skills they had just learned, and 3) a *delayed test session*, approximately 3 days later, to assess retention of these skills.

HyperCard Authoring Tasks

The experiment was conducted using an Apple Macintosh II with an 11-inch monochrome display running the HyperCard application (Apple Computer, 1987). Low-level HyperCard tasks were used for the study. These tasks were selected because they were from a naturalistic setting and could be combined to create (author) a HyperCard stack. Tasks were also chosen to allow us to assess retention of previously learned skills and transfer of training. Tasks for the immediate and delayed test sessions were either: 1) the *same* as tasks from the training session, 2) *similar* to tasks from the training session (e.g., copy field instead of copy button), or 3) *different* than those used in the training session, but the same in the two test sessions. Table 3 presents the three types of tasks. All tasks were counterbalanced between participants and within sessions with a Latin square.

Instructional Materials

Procedural instructions were created for the basic authoring tasks of HyperCard and were presented in two ways. The first was similar to the written procedural instructions found in many online help systems and in textual documentation. An example of the written procedure for copying a HyperCard button is shown in Figure 1. All steps in these written instructions were shown concurrently and filled the Macintosh screen. A time allowance of 4 seconds per procedural step, determined in pre-testing, equalled the total presentation time. The other type

of instruction was a real-time, animated demonstration of the interface procedures generated with the Affinity's® Tempo™ "macro" facility (Affinity Microsystems, Ltd., 1986), but without any accompanying written or spoken text. The animated demonstrations were shown on the display as if another person were performing the steps needed to accomplish an authoring task.

Table 3

Same, Similar, and Different Tasks in the Training, Immediate Test, and Delayed Test Sessions

TRAINING TASKS		IMMEDIATE TEST TASKS		DELAYED TEST TASKS
Copy Text	»»»»»»»»	Copy Text	»»»»»»»»	Copy Text
Link Button	»»»»»»»»	Link Button	»»»»»»»»	Link Button
Modify Field	»»»»»»»»	Modify Field	»»»»»»»»	Modify Field
Create Button	>>>>>>>	Create Field	»»»»»»»»	Create Field
Copy Button	>>>>>>>	Copy Field	»»»»»»»»	Copy Field
Create Card		Delete Card*	»»»»»»»»	Delete Card
Create Stack		Delete Button*	»»»»»»»»	Delete Button

»»»» Task is the **same** between sessions.
>>>> Task is **similar** between sessions except for type of object.
* Task is **different** from previous session.

For each task, equivalencing procedures, similar to those conducted by Baggett (1979), were undertaken in an attempt to assure that differences between instructions were in terms of media rather than content. Four additional participants were given labeled written instructions and asked to look for dissimilarities or missing items in the animated instructions. Based on the equivalence procedures, 6 changes were made to the 45 total instructions in the 7 tasks used during the training session. Most of these changes were in the wording of the written instructions to provide greater consistency between the two media. Appendix A contains the final set of textual instructions for each of the authoring tasks.

1. Select Button Tool from Tools menu.
2. Click on button to copy.
3. Select "Copy Button" from Edit menu.
4. Select "Paste Button" from Edit menu.
5. Click on middle of button and drag to move the button to correct location.
6. Select Browse Tool from Tools menu to determine if field was modified correctly.

Figure 1. *Written procedures for copying a button.*

As shown in Table 4, the presentation times for the two media types were not equivalenced. Four seconds per procedural step equalled the presentation time for the written instructions while the animated demonstrations had longer presentation times (from 6 to 20 seconds total). The longer presentations for the animated demonstrations were the result of additional time required for mouse movements and system response time. Although this was a potential time advantage for the users of the animated demonstrations, we felt that the serial nature of the demonstrations necessitated the additional time. Users of the animations were not allowed to review parts of the demonstrations or return to previous portions of the instruction, whereas users of the written instructions were afforded reading advantages, such as being able to scan and re-read text. Thus, we felt the time allowances made for a fair comparison between the two treatments. In both cases we found through pretesting that the time for the animated and written instructions was adequate for reading and viewing, respectively.

Procedure

The experiment was conducted on two days for each participant, the first lasting approximately 90 minutes and the second lasting about 20 minutes. During the first day each participant received preliminary training on HyperCard, learned authoring tasks using one of the two media in the training session, and performed authoring tasks without any instructions in the immediate test session. Before leaving, participants filled out a questionnaire, and were

asked not to use HyperCard until they had completed the entire experiment. On the second day, 3 to 7 days later, participants received some initial warm-up trials and performed the same authoring tasks as those in the immediate test session.

Table 4

Number of Procedural Steps and Times for Task Instructions for Both Media.

LEARNING TASKS	PROCEDURAL STEPS	TOTAL PRESENTATION TIME (seconds)	
		TEXT	ANIMATION
Create Card	2	8	14.40
Create Stack	4	16	35.50
Copy Text	5	20	30.40
Copy Button	6	24	33.27
Link Button	7	28	40.72
Modify Field	8	32	42.09
Create Button	9	36	46.56
Mean Time	5.86	23.43	34.71

Preliminary HyperCard training. During the first day each participant was familiarized with the basic functions of HyperCard by reading a tutorial (see Appendix B). Included were online skills in how to browse through cards, the basics of how cards are arranged in stacks, and the major components of the system. A short criterion test was administered after the preliminary HyperCard training to insure that the participant had understood the concepts as well as how to navigate through the HyperCard system (see Appendix C). Those areas in which the participant was still deficient were re-taught and tested again. The training period, including criterion test, lasted approximately 30 minutes.

Training trials. For the training tasks shown in Table 3, a goal name, or task, was presented along with a precise description for the goal. For example, the HyperCard display for the task "Copy Button" can be seen in Figure 2. Each participant was told to read the goal and its description thoroughly and then to invoke the instruction for that task. To begin, participants

initiated a Tempo "macro" which began the animated or written instructions. This invocation sequence was practiced in a series of simple non-Hypercard goals before data collection began. Once invoked, the instruction appropriate for the stated task was immediately provided. Participants were not permitted to determine the pace of the instruction or to stop it prematurely.

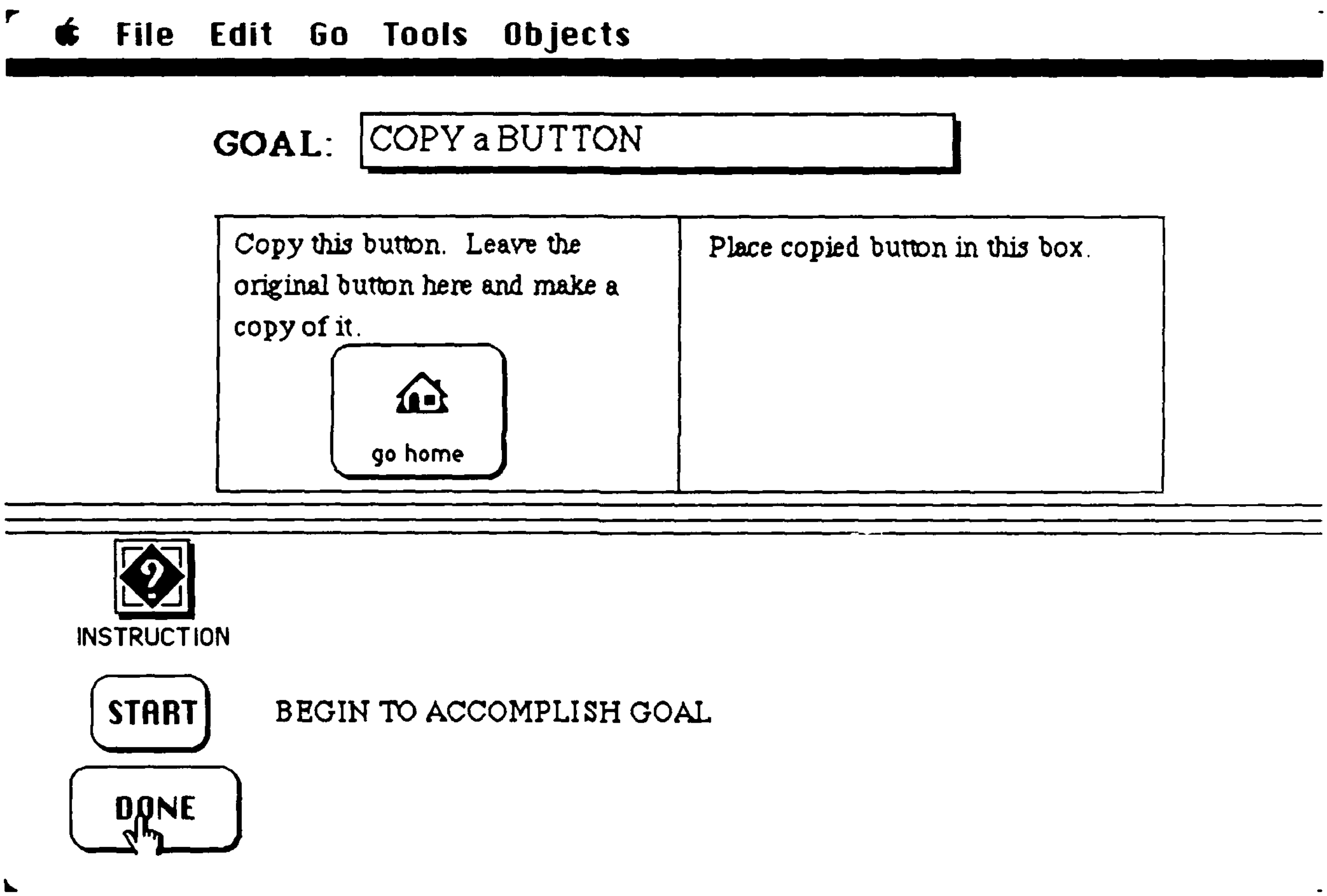


Figure 2. *HyperCard display for learning how to "Copy a Button" during the training session.*

After the textual or animated instructions were presented for a task, participants clicked the mouse on a "start" button, performed the task, and clicked on a "done" button when they were finished (refer to Figure 2). The system then tested the results for correctness and

provided limited feedback about their performance[1] . If incorrect, participants were required to ask for the instructions and to perform the task again until correct. Once a task was completed correctly, participants in the 1-mandatory trial condition continued on to the next task. Those in the 3-mandatory trial condition were required to practice the same task at least three times before moving ahead.

Test sessions. On the immediate and delayed test sessions, participants performed the tasks shown in Table 3 which were the same, similar, and different from the tasks in the training session. In these sessions they were required to perform the tasks without the instructions.

RESULTS

Background

General and Macintosh computing experience was analyzed to see if the random assignment of participants to the experimental treatments (demonstrations vs. text and 1-mandatory vs. 3-mandatory trials) yielded comparable user groups. Independent t-tests comparing levels of experience for each of the measures shown in Table 2 found no significant differences ($p > 0.10$) between the participants receiving the animated demonstrations and those receiving the textual instructions. However, t-tests between the 1-mandatory and 3-mandatory trial groups did reveal significant differences ($p < 0.05$). The 1-mandatory trial group had significantly more experience with computers in classes than the 3-mandatory trial group (2.6 vs. 1.7 courses; $t[13] = -2.18$, $p < 0.05$), a higher subjective rating of programming experience (2.8 vs. 2.1; $t[13] = -2.35$, $p < 0.04$) and a higher subjective

[1] Within the HyperCard system, we were able to examine if the user had executed the essential portions of the task correctly by checking what commands had been used and how the end object appeared. For example, if the user was required to create a new button and put it in a specific location, the user would not be allowed to copy an existing button or to place it incorrectly. When users improperly performed the task, they would be notified of this when signifying that they were done.

rating of their experience with Fortran (3.3 vs. 2.3; $t[13] = -2.19$, $p < 0.05$). These results suggest that comparisons of learning and transfer performance between the 1-mandatory and 3-mandatory trial groups could be influenced by differences in computing experience.

Performance Results on Tasks

In the following sections are the performance results for: 1) tasks that were identical over all three sessions, 2) tasks that were similar over the three sessions, and 3) tasks that were identical over the last two sessions. In each section, the dependent measures will be discussed in the following order: a) *total time* which includes times for both correct and incorrect trials, b) *accuracy* which is the percentage determined by the number of correct trials divided by the number of total trials for each task, and c) *time per attempt* which is the total time divided by number of trials for each task. For brevity, only the statistically significant and meaningful results are reported; however the interested reader can refer to Appendix D for the means of all dependent measures in the different conditions.

Identical tasks. An analysis of variance (ANOVA) on the mean total times to complete the three tasks which were the same in all three performance sessions revealed main effects of task ($F[2,48] = 3.93$, $p < 0.03$) and session ($F[2,48] = 4.92$, $p < 0.01$). The task effect was expected since the tasks were of different length and difficulty. This effect will not be discussed further since there was no interaction of task and medium ($p > 0.5$) to indicate that one of the instructional medium was more or less effective with some of the tasks. However, there was an interaction between medium and session ($F[2,38] = 5.77$, $p < 0.01$) indicating that the instructional medium was differentially effective in the three sessions. Figure 3 shows the means for the two media over the three performance sessions.

As can be seen in Figure 3, users of the written text took more than 50% more time than users of the animated demonstrations during the training session. However, after the training session there was a large decrease in total time for users of the written instructions.

This led to total times for the users of the written instructions which were substantially below those of users of the animated demonstrations in the immediate and delayed testing sessions. To interpret these data a *post hoc* simple main effects analysis was conducted. Over the three sessions, users that received the written instructions improved significantly (F [2,48] = 9.58, $p < 0.001$) whereas there was no corresponding improvement for users of the demonstrations ($p > 0.5$). Further comparisons of the means for the different media during each session revealed no significant ($p > 0.05$) differences. Therefore, the data indicate that users of written instructions improved significantly on the same tasks, while users of the animated demonstrations remained at the same level.

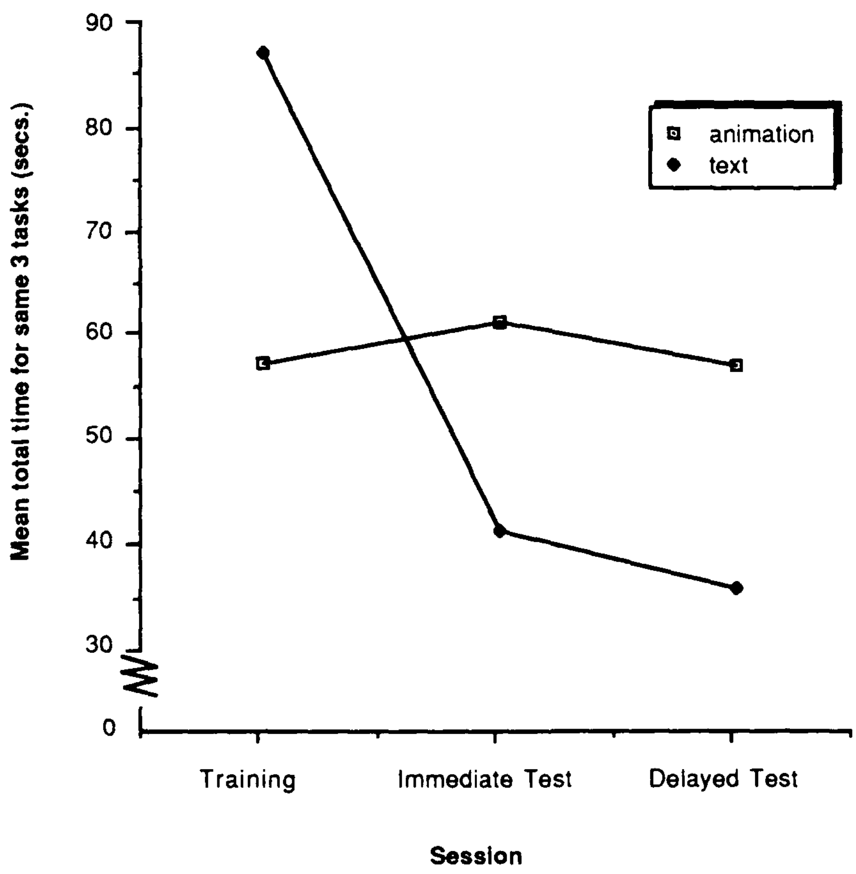


Figure 3. *Mean total time to perform the same three tasks at each performance session.*

To determine whether this improvement from the training session to the immediate test session generally held true for each user of the written instructions, a Fisher sign test was conducted. Twelve of the fourteen users in the text group reduced their total mean time between training and immediate testing sessions (p = 0.0065). The corresponding number of users in the demonstration group who reduced their times between the training and the immediate testing session was eight, (p = 0.395). In fact, the decrease in time between training and immediate testing for the written instructions was the only significant sign test when looking at all possible differences between sessions.

An analysis of the accuracy data followed a pattern similar to the total time data. Main effects of task (F [2,48] = 5.59, $p < 0.007$) and session (F [2,48] = 5.41, $p < 0.008$) were found. Once again, differences between tasks were expected and not of specific interest to this investigation since task and medium did not interact statistically ($p > 0.35$). However, there was an interaction of session and medium (F [2,48] = 5.15, $p < 0.015$) indicating that accuracy of users depended on the performance session. The accuracy data for the interaction of session and medium are shown in Figure 4.

As illustrated in Figure 4, accuracy in performing the tasks was generally high (90% +) with the exception of users in the text group during the first session (79.3%). A *post-hoc* simple main effect comparison found that text group was significantly less accurate than the animation group during the training session. Moreover, over the three test sessions the accuracy for the text group improved significantly (F [2,48] = 7.77, $p < 0.002$) while the animation group did not improve significantly ($p > 0.5$). A Fisher sign test showed that users in the text group generally performed more accurately (11 of 14 subjects) from the training to the immediate test session (p = 0.0287). These results confirm that accuracy did not diminish as speed increased in later sessions.

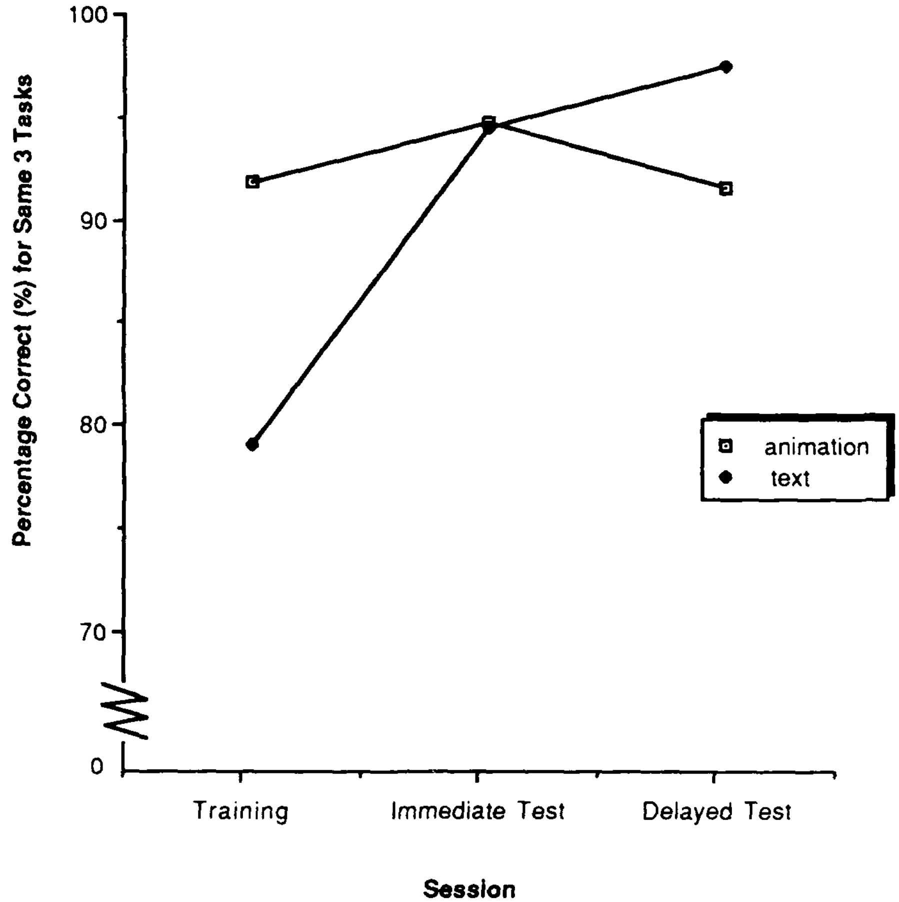


Figure 4. *Mean percentage of correct trials to total trials while practicing the same three tasks at each performance session.*

To check on the rate at which tasks were performed, an ANOVA on the time per attempt was conducted. This analysis found no significant differences between the text and animation groups or in the interaction between medium and sessions ($p > 0.5$). Therefore, all groups during all sessions were performing at approximately the same pace. In total, these results indicated that text users committed more task errors during training, thereby increasing the average number of trials needed to complete training, and ultimately increasing the average total time.

Up to this point little has been mentioned about the effects of the number of mandatory trials on performance. The reasons for this were that statistical analyses were complicated for

one of the dependent measures and all the analyses conducted found no significant differences for this independent variable. For example, a full analysis on total time was not meaningful since the three mandatory trials group obviously would be greater than the one mandatory trial group during the training session. A reduced analyses was conducted to avoid this problem. This analysis was a 4-factor ANOVA with task, session (immediate and delayed test), medium (text and animation), and number of mandatory trials (1 and 3) as the independent variables. However, no significant effects or interactions of mandatory trials were found ($p > 0.05$). The data for 1 or 3 mandatory trials over the three sessions shows this clearly in Figure 5. As illustrated, there is the expected difference between the 1 and 3 mandatory trials during the training session. However, in the later immediate and delayed test session, there is virtually no difference between the groups.

Full analyses on accuracy and time per attempt were possible since each dependent variable was divided by the total number of trials. This manipulation corrected for the number of attempts to complete the task during each session. However, the results of these analyses were similar and revealed no significant effects of mandatory trials or any interaction with any other independent variable ($p > 0.05$). These results were surprising since it was believed that the amount of practice could effect performance differently in the two media. Perhaps the higher level of computing experience for the 1-mandatory trial group could have affected these results. However, based on these data, it appears that more practice does not improve time, accuracy, or rate of performance in any of the conditions.

Similar tasks. To study transfer of training in this study, those tasks which were similar between the training and testing sessions were compared: Copy Button - Copy Field and Create Button - Create Field (see Table 3). The tasks from training to non-training sessions differed only in the object on which they operated. For example, in the training session the task "Copy Button" was changed to "Copy Field" in the later sessions. The operations for these tasks

are very similar except that the change of objects (Button to Field) required a few changes in the procedures (e.g., "Select Field Tool" instead of "Select Button Tool").

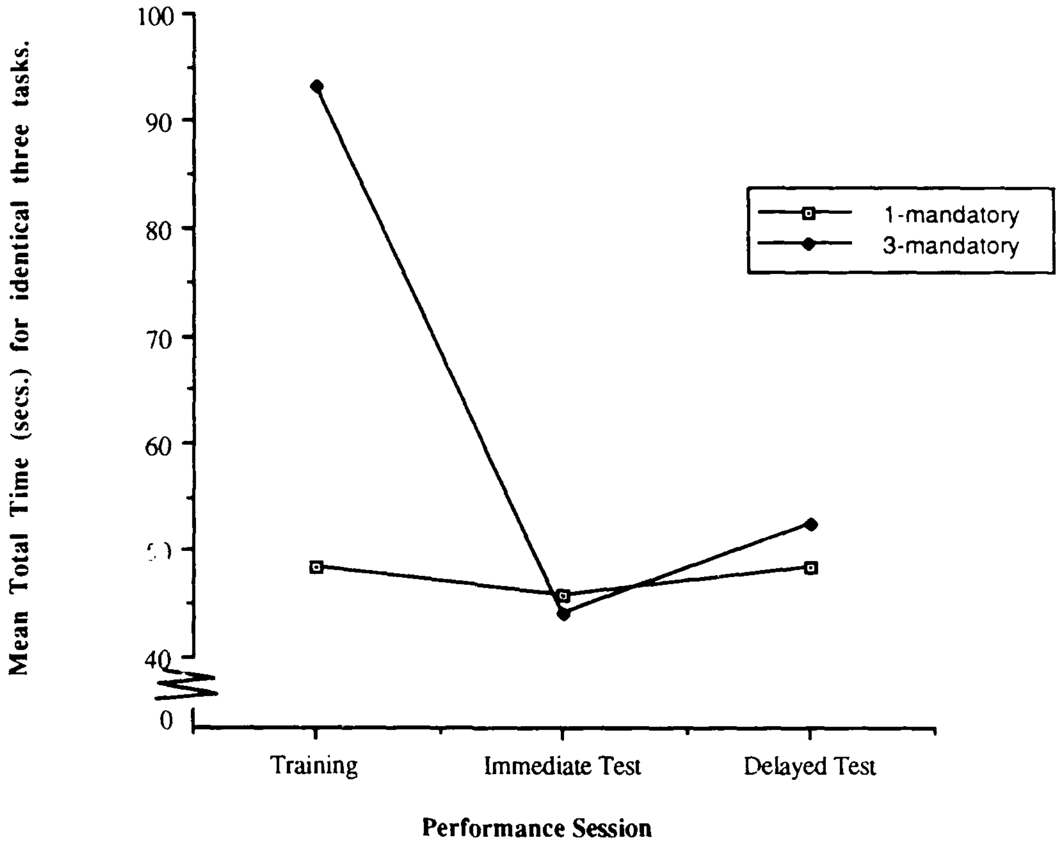


Figure 5. *Mean total time for mandatory trial groups performing 3 identical tasks at each performance session.*

Mean total performance time for each group over the three sessions are seen in Figure 6 and were analyzed using paired t-tests. Between the training session and the immediate test session the users in both groups had a significant change in performance (animation: t [13] = -2.25, $p < 0.045$; text: t [13] = 2.90, $p < 0.015$). The text group participants were able to complete the similar tasks more quickly than the original tasks, whereas the animation group participants took more time when faced with a similar but different task. Although Fisher sign tests for these performance changes were not significant (9 of 14 text users required less time: $p = 0.21$; and 9 of 14 animation users required more time: $p = 0.21$), the results do

suggest that when presented these similar tasks text users experienced positive transfer and animation users experienced negative transfer. After performing these similar tasks in the immediate test session, there was a significant reduction in time to perform these same tasks in the delayed test session for the animation group (t [13] = 3.23, $p < 0.0066$) and no corresponding change in time for the text group ($p > 0.5$). It appears that once the tasks were learned, the animation users were able to perform during the delayed test session at a level similar to the training session (63.9 secs. at training and 63.1 secs. at delayed). This general return to performance similar to that observed in the training session was confirmed by a Fisher sign test. Thirteen of the fourteen users in the animation group reduced their total mean time between immediate and delayed testing sessions ($p = 0.0009$).

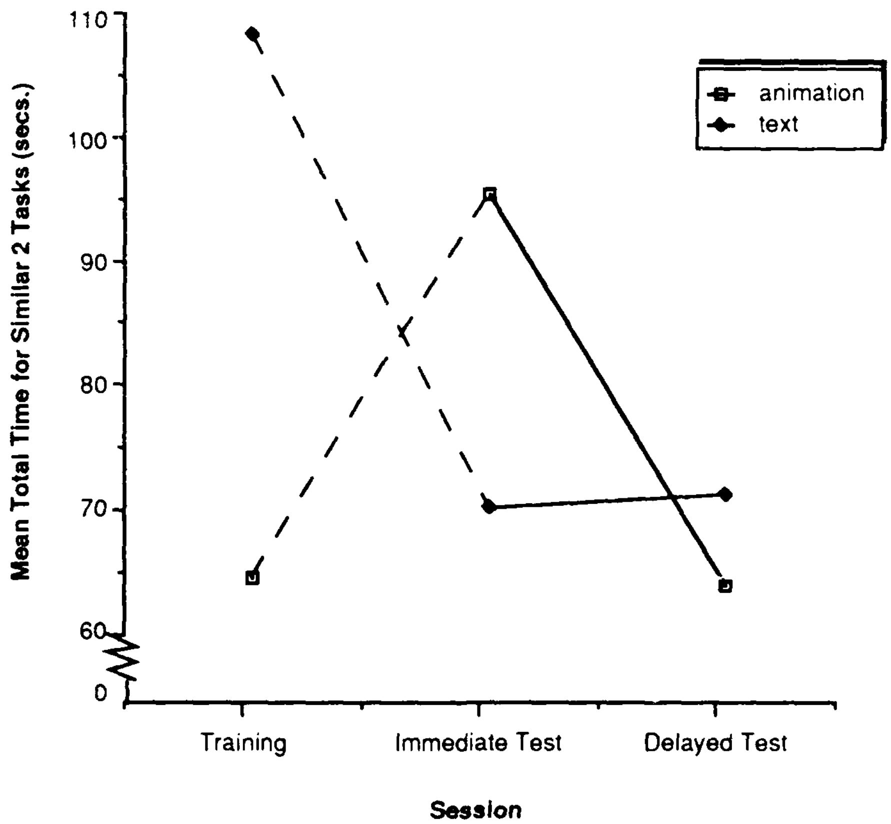


Figure 6. *Mean total time to perform two similar tasks at each performance session.* (dotted lines signify that different, but similar, tasks are compared between the training and later test sessions)

No significant differences were found between the two media on the basis of accuracy between sessions. However, for time per attempt, shown in Figure 7, there was an increase for both media groups between the training and immediate test session (animation: $t\,[13] = -4.74$, $p < 0.0004$; text: $t\,[13] = -2.39$, $p < 0.035$). This result is also generally true for both groups as shown by a Fisher sign test. All participants in the animation group increased their time per attempt ($p < 0.0001$) while thirteen out of fourteen in the text group increased their time per attempt ($p < 0.0009$). In addition, the time per attempt for the animation group decreased between the immediate test session and the delayed test session ($t\,[13] = 5.90$, $p < 0.0001$). These data suggests that the occurrence of a similar, but new task causes users to slow down their rate of performance perhaps requiring more time to problem solve in the immediate session.

Identical tasks over the 2 test sessions. The amount of information which was retained between immediate and delayed test sessions was analyzed by looking at the seven tasks which were the same between the two sessions. To a large extent the results of these analyses were dominated by task effects. For example, highly significant task effects were found for all dependent measures (total time: $F\,[6,144] = 9.14$, $p < 0.0001$; accuracy: $F\,[6,144] = 5.86$, $p < 0.0001$; time per attempt: $F\,[6,144] = 19.79$, $p < 0.0001$). In addition, smaller interaction effects of tasks with other independent variables were found. For the total time measure a third order effect of task, session, and mandatory trial variables was found ($F\,[6,144] = 2.25$, $p < 0.04$) and for the accuracy measure an interaction of task and session was found ($F\,[6,144] = 3.19$, $p < 0.01$). Since none of these effects and interactions involved the type of media (the main focus of the study), further presentation of the data will not be reported. The lack of significant differences due to media on these identical tasks in the test sessions suggests that the type of medium does not affect how well the procedural skills are retained and remembered over a period of time.

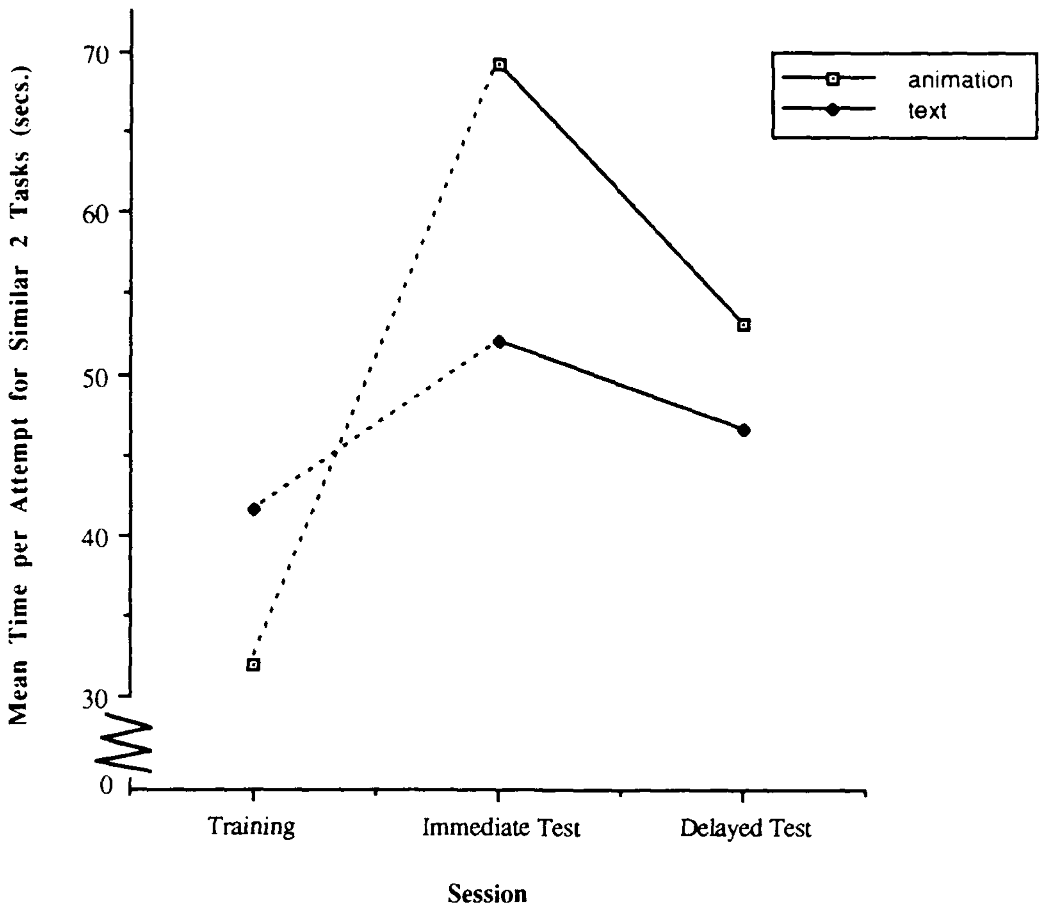


Figure 7. *Mean time per attempt to perform two similar tasks at each performance session*(dotted lines signify that different, but similar, tasks are compared between the training and later test sessions).

Questionnaire Results

The results of the questionnaire administered at the end of the first day of testing can be found in Appendix E. There were no significant differences between the medium groups, however, there were a few significant differences between the two mandatory trial groups. The 1-mandatory trial users wanted the entire set of instructions or demonstration again more than the 3-mandatory trial users (see Question #2; subjective rating of 3.6 vs. 3.5, respectively; $t\,[13] = -3.663$, $p < .0029$). As expected, the 3-mandatory trial participants rarely wanted more practice during the training session, whereas the 1-mandatory trial

participants more often wanted to try a task again (see Question #3; subjective rating of 4.2 vs. 3.0 respectively; t [13] = 3.19, p < .0071).

Interesting trends, although not significant, involving the participant's general opinion of HyperCard (see Questions #8-11) can be summarized as follows: 1) demonstration participants found HyperCard more satisfying than those participants who received textual instructions, 2) participants in the 1-mandatory trial text group found HyperCard most difficult, and 3) 3-mandatory trial participants found HyperCard easier than their counterparts.

DISCUSSION

We expected that animated demonstrations would reduce the amount of translation needed when getting started, serve as an interface example, and provide instructions that are more explicit because they are integrated with the interface. To some extent these predictions were true. The animation group was almost 50% faster than the text group in the training session. Yet, once the instructions were removed the animation group did not improve like the text group did. As we discuss the possible reasons for these contrary results, we will also explore how these animated demonstrations could be improved for advanced learning and retention along with the faster training capability already seen.

The two groups exhibited different behavior which might help explain the unexpected results. Some members of the animation group anecdotally reported that during the training session they had "... just done what the computer had done...". Evidence for this mimicking behavior is also suggest by the data. For the same tasks across sessions performance for the animation group was remarkably stable. In contrast, the text group, although slower and more inaccurate during the training session, improved their performance in the immediate and delayed sessions so that they were approximately 20 seconds faster and 10% more accurate than the animation group. Since performance does not improve for the animation group, they

may have been stuck at the mimicking stage with their representation based solely on a rote procedure.

One reason for this mimicking behavior may be due to the different processing required by text and animation which may account for the different performance results. As discussed earlier, the text group was exposed to a richer encoding medium because it contained verbal, visual, and motoric codes, whereas the animation group had only the visual and motoric components. The text group also had an advantage because the processes involved with reading are familiar to most people since reading is such a common task. In fact, the text group could have visualized the instructions as they read them, providing an even richer visual code during instruction and performance. Other evidence for the different processing of the animated demonstrations was that some participants vocalized what they saw. Users seemed to want a verbal component with the demonstrations suggesting that all three codes (verbal, visual, and motoric) may be desirable. Indeed, having all three codes during learning and performance could lead to a deeper and richer level of processing. This may explain the extra time required for text users during the training session. These users probably needed additional time to translate the verbal instructions into visual and motoric codes. Thus, although reading instructions is commonly disliked and often skipped (Carroll, Mack, Lewis, Grischkcwsky, & Robertson; 1985), the improved performance by the text group suggests that this processing may be beneficial.

Perhaps animated demonstrations and written instructions should be used in combination. In a study by Booher (1975), the relative comprehensibility of various picture-word formats was compared for procedural instructions. The highly pictorial formats which included text were consistently faster and more accurate than the other formats. Using only highly textual formats increased performance time. However, a picture only format increased task errors. The results of the Booher study, which concentrated on static pictures and not animated demonstrations, bear a similarity with the results of this study. Visual instructions

appear to provide the direct mapping from the instruction to the action to be performed. Booher suggests that users pick up advance information from pictures which help them to organize. He goes on to suggest that "...the human processing system is most efficient in comprehension of instructions when the pictorial mode is used to aid in selection and organization of a range of actions and the verbal material is available to confirm specific actions within the range." (p. 268) Thus, it may be that demonstrations with accompanying text would allow for faster learning and improved retention.

The text group was also able to more quickly infer the needed procedures when similar, not identical, tasks were to be performed. It was expected that animated demonstrations would serve as an interface example of a task, making it easier to infer procedures. Yet, the animation group was much slower than the text group at determining the corresponding procedures for a similar task (approximately 20 seconds per task). The slower learning of similar tasks may be due to the inability to remember the task from the training session or the inability to grasp the consistency in methods between the similar tasks. In addition, if participants were only mimicking the training procedures, resulting in a superficial encoding of the task, they were probably unable to alter the procedures from the training task when a similar task was presented in the immediate test session.

These results also call into question the hypothesis that animated demonstrations might serve as interface examples. It has been noted that well developed examples can aid learning (Lewis & Anderson, 1985; Sweller and Cooper, 1985). However, the degraded performance of animation users between the training and immediate test session for similar tasks suggests that these users did not view the training tasks as generalizable examples. Instructions given to the participants before the training session included a warning that they would see tasks that were "... the same or similar to the tasks you will learn about...". Yet, they seem to have seen each task's procedure as a solitary method to accomplish a single goal without looking for consistencies between the training tasks and similar immediate test tasks. Perhaps having only

one task as an "example" is not enough of a base for making inferences. In reading tasks involving problem solving, Black, Kay, and Soloway (1987) suggest that two exemplar stories are needed in different contexts before people can see a generalized plan. Perhaps adding a second training task, which is similar to the immediate test task, would have helped the animated demonstration group in determining the necessary procedures. Still, it is important to remember that the differences can not be totally explained because the text group also had only a single "example" and, yet, were able to more quickly see consistencies in the methods needed to perform the similar, but new, tasks.

In addition, the results of Lewis, et al. (1987), where users were able to infer relationships between commands and display changes, would suggest that users are able to understand unstated relationships. But those examples were much simpler than the relationships used in this study. Animated demonstration users had to extract the necessary steps out of a long procedure in order to create a new series of steps. Perhaps if the animated demonstrations are broken down into smaller segments, as they were presented in the Lewis, et al. study, users would be able to use individual segments to create a new method. Interestingly, this breakdown of procedures into elemental steps is currently used in Apple's "Guided Tours" for the Macintosh. Having single chunks of a procedure may more easily enable users to piece together more complex procedures.

This observation leads us to the final explanation for our results. The type of animated demonstrations used in this study may be the reason that the demonstrations did not perform as well as expected. The amount of time given to the animation group to watch the training sequences was determined somewhat arbitrarily. This time constraint, combined with the user's inability to stop the demonstration or begin a demonstration in the middle, may have led to poorer performance. Although the text group was also given a time constraint, they were able to read any part of the written instructions and at any speed. In addition, the structure of some tasks may not be suitable for animated demonstrations. If a task is a combination of lower

level tasks, the task structure may not be apparent if the subtasks are simply strung together in a serial demonstration. In addition, human capabilities such as memory span and eye fixations, should be considered when determining the framework for the animations.

CONCLUSION

Animated demonstrations will not be an appropriate online help aid for all interfaces. Their use is limited to interfaces which are highly graphical and contain few hidden responses to user input. Yet, this technique is being used to teach procedural methods in direct manipulation interfaces such as the Macintosh and CAD/CAM systems. And, although demonstrated procedural instructions might appear to have a clear advantage over other, less high-tech media such as written instructions, it could not be deduced from this study. Thus, the way in the which the animated demonstrations are constructed should be carefully undertaken to complement the task and human capabilities. Animated demonstrations, as were implemented for this study, did have an advantage over reading because they speeded up the getting started process. Unfortunately, they did not require the processing which would aid in future learning. More research into ways of combining the ease of learning with later retention and problem solving could enhance this instructional technique for direct manipulation interfaces.

ACKNOWLEDGEMENTS

This work was supported under ONR Contract number N00014-87-K-0740 with John J. O'Hare serving as the technical monitor. The authors wish to thank Richard Gong for his help in data collection and for his insights. The conclusions within this paper are those of the authors and not necessarily those of ONR.

REFERENCES

Affinity Microsystems, Ltd. (1986). *Tempo manual.* Boulder, CO.

Anderson, J. R., Boyle, C. F., Farrell, R., & Reisner, B. J. (1984). *Cognitive principles in the design of computer tutors.* (Technical Report #ONR-84-1). Pittsburgh: Carnegie-Mellon University, Department of Psychology.

Apple Computer, Inc. (1987). *HyperCard manual.* Cupertino, CA.

Apple Computer, Inc. (1986). *Macintosh II manual.* Cupertino, CA.

Baggett, P. (1979). Structurally equivalent stories in movie and text and the effect of the medium on recall. *Journal of Verbal Learning and Verbal Behavior, 18,* 333-356.

Baggett, P. (1987). Learning a procedure from multimedia instructions: The effects of film and practice. *Applied Cognitive Psychology, 1,* 183-197.

Black, J. B., Kay, D. S., & Soloway, E. M. (1987). Goal and plan knowledge representations: From stories to text editors and programs. In J. M. Carroll (Ed.) *Interfacing thought* (pp. 36-60). Cambridge, MA: MIT Press.

Booher, H. R. (1975). Relative comprehensibility of pictorial information and printed words in proceduralized instructions. *Human Factors, 17* (3), 266-277.

Carroll, J. M., Mack, R. L., Lewis, C. H., Grischkowsky, N. L., & Robertson, S. R.. (1985). Exploring exploring a word processor. *Human-Computer Interaction, 1,* 284-307.

Carroll, J. M. and Mazur, S. A. (1986). LisaLearning. *Computer, 19* (11), 35-49.

Cullingford, R. E., Krueger, M. W., Selfridge, M., and Bienkowski, M. A. (1982). Automated explanations as a component of a computer-aided design system. *IEEE Transactions on Systems, Man, and Cybernetics, 12* (2), 168-181.

Duisberg, R. A. (1988). Animation using temporal constraints: An overview of the Animus system. *Human-Computer Interaction, 3,* 275-307.

Grignetti, M. C., Hausmann, C., and Gould, L. (1975). An "intelligent" on-line assistant and tutor - NLS-SCHOLAR. In *Proceedings of the 1975 National Computer Conference, 44,* (pp. 775-781), Anaheim, CA: AFIPS Press.

LeFevre, J. A. and Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction, 3* (1), 1-30.

Lewis, C., Casner, S., Schoenberg, V., and Blake, M. (1987). Analysis-based learning in human-computer interaction. In *Proceedings of INTERACT '87* (pp. 275-280). New York: IFIP.

Lewis, M. W., and Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. *Cognitive Psychology, 17,* 26-65.

Mack, R. L., Lewis, C., and Carroll, J. M. (1983). Learning to use a word-processor: Problems and prospects. *ACM Transactions on Office Information Systems, 1,* 254-271.

MacroMind, Inc. (1987). *VideoWorks II. Animated presentations for the office, school or home*. Chicago, Ill.

Myers, B. A. (1987). Creating dynamic interaction techniques by demonstration. In *Proceedings of CHI+GI 1987* (pp. 271-278). New York: ACM.

Neiman, D. (1982). Graphical Animation from Knowledge. In *Proceedings of National Conference on AI,* (pp. 373-376). Menlo Park, CA: AAAI.

Paivio, A. (1971). *Imagery and verbal processes.* NY: Holt, Rinehart, & Winston.

Sweller, J. and Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2* (1), 59-89.

APPENDIX A: Written Procedures for the Authoring Tasks

<u>Written procedures for creating a new card.</u>
1. Select "New Card" from Edit menu.
2. Select "Prev" from Go menu to insure card has been placed correctly.

<u>Written procedures for creating a new stack.</u>
1. Select "New stack..." from File menu.
2. Type in name for stack ("My Stack").
3. Click on "New" to create new stack and leave stack creation window.
4. Select "Back" from Go menu to return to previous stack to insure you have created a stack.

<u>Written procedures for copying text.</u>
1. Select Browse Tool from Tools menu.
2. Select text from field which you want to copy.
3. Select "Copy Text" from Edit menu.
4. Click on location in another field where copied text should be placed.
5. Select "Paste Text" from Edit menu.

<u>Written procedures for copying a button.</u>
1. Select Button Tool from Tools menu.
2. Click on button to copy.
3. Select "Copy Button" from Edit menu.
4. Select "Paste Button" from Edit menu.
5. Click on middle of button and drag to move the button to correct location.
6. Select Browse Tool from Tools menu to verify if button has been created correctly.

<u>Written procedures for linking a button.</u>
1. Select Button Tool from Tools menu.
2. Click on button to be linked.
3. Select "Button Info..." from Objects menu.
4. Click on "Link to..." from Button Info window.
5. Go to stack which you want button to link to (Home stack).
6. Click on "This Stack" button in linking window.
7. Select Browse Tool from Tools menu.

<u>Written procedures for modifying a field.</u>
1. Select Field Tool from Tools menu.
2. Click on field to modify.
3. Select "Field Info..." from Objects menu.
4. Click on desired style for field (scrolling).
5. Click on "Font..." in Field Info window.
6. Select desired Font (Times) and Point Size (12).
7. Click on "OK" to select font and exit Font window.
8. Select Browse Tool from Tools menu to determine if field was modified correctly.

<u>Written procedures for creating a button.</u>
1. Select "New Button" from Objects menu.
2. Select "Button Info..." from Objects menu.
3. Select style for Button (transparent)
4. Select if Show name and Auto hilite should be selected.

5. Click on "Icon..." button in Button Info window.
6. Select desired icon for the button (Art Ideas) and click on "OK" to exit Icon menu.
7. Click on middle of button and drag to move button to location.
8. Click on corner of the button and drag to resize the button.
9. Select the Browse Tool from Tools menu to determine if button has been created correctly.

APPENDIX B: Training Tutorial

Goals and Overview of the Experiment

The experiment today is hopefully going to help us discover more about how people learn to use software, and in particular, how people use HyperCard. In addition, you will learn about the new Macintosh application called HyperCard. Hopefully, you will be able to use the information you learn today if you want to use HyperCard in the future. As we go through the next two hours be sure to stop the experimenter if you don't understand some instruction given to you.

The experiment will be divided into 3 major portions. First you will read some training about the basics of HyperCard, called "Browsing." You will need to be able to understand these basics in order to work on the next part of your task, which will be the "Authoring" portion of HyperCard. After the training, you will work on tasks which require you to do "Authoring". After these tasks, there will be a post-test period which will allow us to find out how much information you remember and will even expose you to some things you may not have learned.

HyperCard Browsing Training

Instructions for Training

As you go through this self-guided training you will be given information about HyperCard as well as guides of how to use HyperCard. Whenever you see something like this:

==> **DO** something

it means that the following information will be an instruction to look at the Macintosh screen or to do something. It is very important that you follow each of these steps so that you can gain some experience with using the interface of HyperCard.

Please, read each of the following sections carefully! You will be given a review test covering the training material at the end of the training and you will be given a chance to review any of the things you missed.

What is HyperCard?

You may be wondering what HyperCard really is. HyperCard is an information organizer in much the same way as a library card catalog organizes the books in a library. It allows you to access information in many fast and convenient ways. HyperCard has been made flexible enough, though, that you can organize information in a way that you want it to be organized. Secondly, HyperCard is an interface creation tool. HyperCard provides the capability of making your own interface. This capability is called the AUTHORING portion of HyperCard. The majority of your time today will be spent in learning about AUTHORING.

Basic Elements of HyperCard

There are some basic elements of HyperCard that you need to understand to allow you to work with HyperCard. As you go through each of the concepts, they will be related to the real-world environment of a library card catalog.

Stacks

A STACK is a homogeneous collection of information in the form of CARDS. A stack in HyperCard is like a drawer in a card catalog, but it contains only one topic area. For example, imagine a card catalog with information about only biology or chemistry -- this would be the same information you would find in a stack.

Each stack is stored in a separate Macintosh disk file.

==> **Look** at the card you see on the screen. This is a card in a stack. The stack is the Home stack. On the card there are some icons which represent some other stacks.

You will be looking at the Phone Directory stack to show you what a common stack might look like.

==> **Open** the Phone Directory stack by clicking once on the icon of the Phone Directory stack.

Phone Directory

Cards

A CARD is an element in a stack which may contain text, graphics, or both. A card in a stack is like a card in a card catalog which contains information related to the content of the card catalog.

==> **Look** at the screen. You are now looking at a card. It is the contents of ENTIRE screen except for the menubar (the bar at the top of the screen with the menus File, Edit, Go etc.), not just the image of the Rolodex card itself.

The cards in a stack are set up in a circular fashion -- when you reach the last card in a stack, the next card will be the first card of the stack.

==> **Click** on the right arrow at the bottom of the screen. You will go the next card in the stack. Keep clicking on the right arrow. You will notice that you will return to the first card of the stack, the card for Adams, because the stack is linked in a circular manner. Make sure to end up on the Adams card after moving through the cards in this stack.

A new card in a stack is created by selecting NEW CARD from the Edit menu. When you ask for a new card, the new card will be inserted after the card you were in.

Backgrounds

A BACKGROUND is the place in a stack where things reside which will appear on every card in the stack. The background of a stack is like the pre-printed matter on every card of a card catalog. For example, in a card catalog, the cards might have a pre-printed border and the name of the library on every card, but that's not the important information on the card.

In HyperCard, there is usually only one background per stack. So, each card in a stack has a background, which is usually the same for all the cards in the stack. In addition to the background, a card has information which is solely unique to the card. To see the background for a card in a stack, select BACKGROUND from the Edit menu.

You will know you are looking at the background because hash marks appear on the menu bar. When the background has been selected, you can see just those things that appear on every card. These are the things that are contained in the background.

==> **Show** the Background of the current card, by selecting BACKGROUND from the Edit menu.

==> **Look** at the menu bar and notice the hash marks. Also, notice that there are some parts of the card, such as the text, that are now missing.

==> **Return** to the card by selecting BACKGROUND from the Edit menu.

Fields

The FIELDS on a card or in a background are the places for textual information which will change from card to card. The field in HyperCard is like the title of a book or a call number on a card in a card catalog.

When you enter a field which can be edited, the BROWSE TOOL (the cursor which looks like a little hand with a pointed finger) becomes an I-beam cursor. You must click in the field to be able to begin typing.

==> **Move** the browse tool (the little hand with a pointed finger) to the area underneath the name and address. The I-beam should appear when the cursor is over the field. Click at the location in the field so that you can begin typing. Type any phone number underneath the name and address for Adams.

Buttons

The BUTTONS of a card or background are activation tools which cause something to happen. Usually buttons are the navigational tools, or links, between cards and stacks. There is nothing to compare to a button in

HyperCard in a card catalog system. It is like a MAGIC button on a card in a card catalog which would link you to an item referenced on the card. For example, if the card had a reference to another card with similar information and you were to push on that place on the card, the card catalog drawer with the referenced card would fly open and the card would open up.

In HyperCard, a button is activated by a single click of the mouse when the browse tool is over the button. Clicking on a button will usually cause something to happen. The right and left arrow buttons, which you have already used, are activation buttons to move you forward and backward through the stack.

==> **Click** on the button on this card which looks like a calendar (top right hand corner of the card). You will then go to the calendar stack and see a calendar.

==> **Return** to the phone directory after your are done looking at the calendar, by clicking on the phone button (an icon for the phone directory).

Sometimes buttons will also do things like sort cards or quickly go through them.

==> **Look** at the 2 buttons in the bottom left corner of this card to see some of these types of buttons.

<u>The Home Stack</u>
An important stack in HyperCard is the HOME STACK. The Home stack is usually accessible by a button with a house on it.

==> **Look** at screen and find the button with a house on it. Click on the house and you will go to the home stack.

The Home stack contains the user's links to other stacks as you can see here on the first card of the stack. Other cards in the Home stack contain information for HyperCard about where to look for stacks, applications, and documents. In addition, there is a card which contains information about the level of complexity which HyperCard should show to a user. You won't need to use these cards today.

<u>Recent</u>
HyperCard has a way of keeping track of the last 42 cards that the user has visited. This ability is like marking the last 42 cards that have been seen in a card catalog no matter where tney are. This information can be accessed by selecting RECENT from the Go menu. You will be shown unique, miniature representations of the cards you have accessed. If a card has been accessed many times, it will only have one representation. If you click on one of these representative cards, you will be sent to that card. The current card has a thick white rectangle around it.

==> **Select** RECENT from the Go menu. Choose the card which is to the right of the first card in the recent screen. You should now be back at the first card of the phone directory.

The Recent capability in HyperCard gives you direct access to a card you have visited before.

Saving in HyperCard

Something that is different between other Macintosh applications and HyperCard is that you don't ever have to explicitly SAVE anything in HyperCard. HyperCard periodically saves changes to a stack automatically. So, during the training and the experiment, you will not need to explicitly ask HyperCard to save anything.

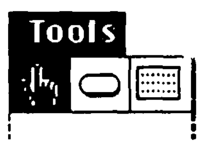
Navigation through Stacks and Cards

==> **Go** home to begin this portion.

Browser Tool

Now you will be given instructions about how to navigate through a stack and between cards. First you need to know about the BROWSER TOOL. The browser tool must first be selected before you use buttons -- which is the usual way of navigation. To select the Browser tool you can select it from the Tool menu.

==> **Go** into the Tools menu and select the browser tool, which should already be selected.



Opening a stack

When you are in HyperCard, to open a stack you click once on a button icon for that stack.

==> **Click** once on the phone directory button again to go to the phone directory stack.

Using Buttons

Normally cards in a stack will have buttons which allow you to navigate through the stack. You simply click once on the button to activate the button. You have already done this in the phone directory.

Using the Go menu

You can also use the Go menu to navigate to important stacks (like the Home stack and the Help stack) or to other cards in the current stack.

==> **Select** NEXT from the Go menu and you will go to the next card.

==> **Select** HOME from the Go menu and you will return to the Home stack.

==> **Select** BACK from the Go menu and you will return to your previous card.

==> **Select** HOME again from the Go menu.

These are the basic ways to navigate within and between stacks.

Now you are done with the Browsing training for HyperCard. You will now be given a short review of the actions and concepts you will need to understand to move on to the next stage of the experiment.

APPENDIX C: Training Criterion Test

HyperCard Browsing Test
During the browsing test, give immediate feedback if right or wrong, with correction if wrong. After you give the subject any needed correction make sure they perform the action correctly before moving on to the next item.

	OK	OK (2nd time)
Please open the To Do Stack (should click on the To Do Stack button)	_____	_____
Can you show me the easiest way to move to the next card in this stack? (should use forward arrow button)	_____	_____
Now can you return to the card you just left? (should only use backward arrow button)	_____	_____
Show me how to go back to home? (should click on the home button or select HOME from the Go menu)	_____	_____
Show me how to go back to the To Do stack - without clicking on the To Do stack icon? (should select BACK from the Go menu or select RECENT from the Go menu - if the subject selects RECENT remind them of the BACK command)	_____	_____
Can you show me what the background of the To Do stack looks like? (should select Background from the Edit menu)	_____	_____
Now let's return to the entire card. (should select Background from the Edit menu)	_____	_____
How do you save things in HyperCard? (should say that you don't need to)	_____	_____
What is another way to go back to the home card - other than the way you went Home before? (should click on the home button or select HOME from the Go menu)	_____	_____

DEPENDENT MEASURE: TOTAL TIME (secs.)
TASK COMBINATION: 3 IDENTICAL TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy text	51.37	19.82	19.70	24.05	18.89	61.96	75.95	30.63	29.33	25.42	24.02	78.21
Link button	85.52	47.20	79.00	102.08	97.64	36.83	153.45	85.73	27.11	58.71	35.24	34.39
Modify field	93.88	37.47	75.08	33.03	107.14	44.86	99.07	70.46	34.59	30.57	31.92	64.30
Mand. Means	76.92	34.83	57.93	53.05	74.56	47.88	109.49	62.27	30.34	38.23	30.39	58.97
Session Means	55.88		55.49		61.22		85.88		34.29		44.68	
Media Means			57.53						54.95			

DEPENDENT MEASURE: TOTAL TIME (secs.)
TASK COMBINATION: 2 SIMILAR TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy butt/fld	53.86	33.27	35.44	42.89	87.96	55.86	74.17	37.94	26.88	46.04	59.46	27.08
Create butt/fld	106.54	61.73	85.61	88.57	114.19	120.21	170.65	147.05	70.48	138.83	103.03	87.69
Mand. Means	80.2	45.0	60.53	65.73	101.08	88.04	122.41	92.50	48.68	92.44	81.25	57.39
Session Means	62.6		63.13		94.56		107.46		70.56		69.32	
Media Means			73.43						82.45			

DEPENDENT MEASURE: TOTAL TIME (secs.)
TASK COMBINATION: 7 IDENTICAL TASKS OVER TESTING SESSIONS

	TEXT				ANIMATION			
	IMMEDIATE		DELAYED		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Create card	17.63	42.35	46.91	41.99	17.86	19.68	23.4	32.83
Create stack	16.81	21.58	18.29	34.07	15.99	26.14	41.15	16.76
Copy text	19.70	24.05	18.89	61.96	29.33	25.42	24.02	78.21
Copy field	35.44	42.89	87.96	55.86	26.88	46.04	59.46	27.08
Link button	79.00	102.08	97.64	36.83	27.11	58.71	35.24	34.39
Modify field	75.08	33.03	107.14	44.86	34.59	30.57	31.92	64.30
Create field	85.61	88.57	114.19	120.21	70.48	138.83	103.03	87.69
Mand. Means	47.04	50.69	70.15	56.54	31.75	49.34	45.46	44.47
Session Means	48.88		63.35		40.55		44.97	
Media Means	56.12				42.76			

DEPENDENT MEASURE: PERCENT CORRECT (%)
TASK COMBINATION: 3 IDENTICAL TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy text	87.9	85.5	78.6	92.9	100.0	91.5	100.0	95.3	88.1	95.3	100.0	100.0
Link button	60.5	61.2	100.0	100.0	88.5	100.0	81.7	91.0	100.0	88.1	73.5	81.3
Modify field	88.3	92.2	92.9	100.0	100.0	100.0	93.5	90.3	100.0	89.3	100.0	93.4
Mand. Means	78.9	79.63	90.5	97.63	96.17	97.17	91.73	92.2	96.03	90.9	91.17	91.57
Session Means	79.27		94.07		96.67		91.97		93.47		91.37	
Media Means	73.43						82.45					

DEPENDENT MEASURE: PERCENT CORRECT (%)
TASK COMBINATION: 2 SIMILAR TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy butt/fld	94.0	90.3	100.0	83.3	100.0	100.0	95.0	100.0	98.3	100.0	100.0	83.3
Create butt/fld	49.3	57.7	92.9	69.0	62.9	78.8	86.2	96.7	92.9	76.2	88.0	79.2
Mand. Means	71.65	74.0	96.45	76.15	81.45	89.4	90.6	98.35	95.6	88.1	99.0	81.25
Session Means	72.83		86.3		85.43		94.48		91.85		90.13	
Media Means			73.43						82.45			

DEPENDENT MEASURE: PERCENT CORRECT (%)
TASK COMBINATION: 7 IDENTICAL TASKS OVER TESTING SESSIONS

	TEXT				ANIMATION			
	IMMEDIATE		DELAYED		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Create card	92.9	92.9	100.0	100.0	89.3	90.5	100.0	100.0
Create stack	100.0	100.0	100.0	100.0	100.0	100.0	88.9	100.0
Copy text	78.6	92.9	100.0	91.5	88.1	95.3	100.0	100.0
Copy field	100.0	83.3	100.0	100.0	98.3	100.0	100.0	83.3
Link button	100.0	100.0	88.5	100.0	100.0	88.1	73.5	81.3
Modify field	92.9	100.0	100.0	100.0	100.0	89.3	100.0	93.4
Create field	92.9	69.0	62.9	78.8	92.9	76.2	88.0	79.2
Mand. Means	93.9	91.16	93.06	95.76	95.51	91.34	92.91	91.03
Session Means	92.53		94.41		93.43		91.97	
Media Means		56.12				42.76		

DEPENDENT MEASURE: TIME PER ATTEMPT (secs.)
TASK COMBINATION: 3 IDENTICAL TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy text	23.54	25.32	30.33	18.04	25.42	23.83	19.82	17.12	30.49	15.32	24.05	19.70
Link button	38.23	37.93	34.39	35.24	36.62	27.11	33.58	28.51	36.83	34.75	46.54	33.86
Modify field	50.25	33.02	58.91	31.92	30.57	34.59	32.12	31.29	44.86	59.62	33.31	71.31
Mand. Means	37.34	32.09	41.21	31.73	30.87	28.51	28.62	25.64	37.39	36.56	34.63	41.62
Session Means	34.72		36.29		29.69		27.13		36.98		38.13	
Media Means	33.57						34.08					

DEPENDENT MEASURE: TIME PER ATTEMPT (secs.)
TASK COMBINATION: 2 SIMILAR TASKS

	TEXT						ANIMATION					
	LEARNING		IMMEDIATE		DELAYED		LEARNING		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Copy butt/fld	32.90	24.73	27.08	42.82	46.04	26.88	27.36	17.96	28.41	62.67	42.89	35.44
Create butt/fld	51.51	56.88	75.15	63.0	62.06	51.24	46.89	35.51	110.95	75.01	69.22	64.67
Mand. Means	42.21	40.81	51.12	52.91	54.05	39.06	37.13	26.74	39.68	68.84	56.06	50.06
Session Means	41.51		52.01		46.56		31.93		69.26		53.06	
Media Means	46.69						51.42					

DEPENDENT MEASURE: TIME PER ATTEMPT (secs.)
TASK COMBINATION: 7 IDENTICAL TASKS OVER TESTING SESSIONS

	TEXT				ANIMATION			
	IMMEDIATE		DELAYED		IMMEDIATE		DELAYED	
	1mand	3mand	1mand	3mand	1mand	3mand	1mand	3mand
Create card	28.69	20.83	19.68	17.86	27.30	28.27	42.35	17.63
Create stack	16.76	41.15	20.83	15.99	34.07	18.29	21.58	16.81
Copy text	30.33	18.04	25.42	23.83	30.49	15.32	24.05	19.70
Copy field	27.08	42.82	46.04	26.88	28.41	62.67	42.89	35.44
Link button	34.39	35.24	36.62	27.11	36.83	34.75	46.54	33.86
Modify field	58.91	31.92	30.57	34.59	44.86	59.62	33.31	71.31
Create field	75.15	63.0	62.06	51.24	110.95	75.01	69.22	64.67
Mand. Means	38.76	36.14	34.46	28.21	44.70	41.99	39.99	37.06
Session Means	37.45		31.38		43.35		38.53	
Media Means	34.42				40.94			

APPENDIX E: Questionnaire Results

Mean responses to Questionnaire for both media types

Question	Demo	Text	p-value
1. How often did you want to see only a portion of the demonstration before you had ever seen it for a specific goal? (1=often, 5=never)	3.79(1.19)	3.36 (.93)	.189
2. How often did you want to see only a portion of the demonstration after you had seen the first presentation for a specific goal? (1=often, 5=never)	3.43(1.53)	1.93 (1.0)	.125
3. When you were working on the tasks in which instructions were provided, how often did you want more practice on a task before going on to the next task? (1=often, 5=never)	3.31(1.07)	3.5 (1.22)	.640
4. Would you have preferred to have the same information as the step-by-step written instructions /demonstration contained in a demonstration so you could watch the procedures being performed/written format which would be accessed online? (1=not at all, 5=very much)	2.86(1.66)	3.36 (1.01)	.405
5. Would you have preferred to have the same information as the step-by-step instructions/ demonstration contained in a written format on paper? (1=not at all, 5=very much)	3.0 (1.84)	2.79 (1.37)	.724
6. How much information was missing from the step-by-step instructions/demonstrations? (1=none, 5=a lot)	1.86(.86)	1.86 (.86)	•
7. How much of the information that you learned when instructions were provided did you use when you were working on the tasks in which instructions were not provided? (1=none, 5=a lot)	4.14(.66)	4.14 (.73)	.612
GENERAL IMPRESSIONS of HYPERCARD			
8. (1=terrible, 5=wonderful)	3.86 (.86)	3.79 (.58)	.775
9. (1=frustrating, 5=satisfying)	3.79 (.68)	3.5 (.52)	.435
10. (1=difficult, 5=easy)	4.21 (.58)	4.0 (.68)	.190
11. (1=dull, 5=stimulating)	3.64 (1.01)	4.14 (.86)	.110

Mean responses to Questionnaire for both mandatory trial groups

Question	1-mandatory trials	3-mandatory trials	p-value
1. How often did you want to see only a portion of the demonstration before you had ever seen it for a specific goal? (1=often, 5=never)	3.64(1.08)	3.5 (1.09)	.612
2. How often did you want to see only a portion of the demonstration after you had seen the first presentation for a specific goal? (1=often, 5=never)	2.93 (1.21)	1.79 (1.25)	.003
3. When you were working on the tasks in which instructions were provided, how often did you want more practice on a task before going on to the next task? (1=often, 5=never)	3.0 (1.11)	4.21 (.80)	.007
4. Would you have preferred to have the same information as the step-by-step written instructions /demonstration contained in a demonstration so you could watch the procedures being performed/written format which would be accessed online? (1=not at all, 5=very much)	3.0 (1.41)	3.21 (1.37)	.711
5. Would you have preferred to have the same information as the step-by-step instructions/ demonstration contained in a written format on paper? (1=not at all, 5=very much)	2.57 (1.60)	3.21 (1.58)	.264
6. How much information was missing from the step-by-step instructions/demonstrations? (1=none, 5=a lot)	1.64 (.75)	2.07 (.92)	.212
7. How much of the information that you learned when instructions were provided did you use when you were working on the tasks in which instructions were not provided? (1=none, 5=a lot)	4.14 (.66)	4.29 (.73)	.635
<u>GENERAL IMPRESSIONS of HYPERCARD</u>			
8. (1=terrible, 5=wonderful)	3.86 (.77)	3.79 (.70)	.775
9. (1=frustrating, 5=satisfying)	3.71 (.47)	3.57 (.51)	.104
10. (1=difficult, 5=easy)	3.93 (.62)	4.29 (.61)	.189
11. (1=dull, 5=stimulating)	4.14 (.86)	3.64 (1.01)	.110

Distribution List for This Report

OSD

Dr. Earl Alluisi
Office of the Deputy Under Secretary
of Defense
OUSDRE (E & LS)
Pentagon, Room 3D129
Washington, DC 20301

U.S. Naval Test Center
Aricrew Systems Branch
Systems Engineering Test Directorate
Patuxent River, MD 20670

DEPARTMENT OF THE NAVY

Cdr. Robert C. Carter USN
Naval Research Laboratory
Head, HCI Laboratory
Washington, DC 20375-5000

Dr. L. Chmura
Computer Sciences & Systems
Code 5592
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 North Quincy Street
Arlington, VA 22217-5000

Commanding Officer
Navy Personnel R&D Center
San Diego, CA 92152-6800

Commanding Officer
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, DC 20361

Commanding Officer
Naval Biodynamics Lab
Michoud Station
Box 29407
New Orleans, LA 70819

Commanding Officer
Naval Weapons Center
Human Factors Branch, Code 3152
Naval Weapons Center
China Lake, CA 93555

Dean of the Academic Departments
U.S. Naval Academy
Crew Station Design
Annapolis, MD 21402-5018

Director
Technical Information Division
Code 2627
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Robert A. Fleming
Naval Ocean Systems Center
Human Factors Support Group
1411 South Fern Street
Arlington, VA 22202-2896

Dr. Sherman Gee
Office of Naval Technology
Command & Control Technology
CODE 221
800 N. Quincy Street
Arlington, VA 22217-5000

Mr. Jeff Grossman
Naval Ocean Systems Center
Code 4403, Bldg. 334
San Diego, CA 92152-6800

Human Factors Engineering
Code 441
San Diego, CA
Naval Ocean Systems Center

Capt. Thomas E. Jones, MSC, USN
Aviation Medicine &
Human Performance (Code 404)
Naval Medical R&D Com
National Capital Region
Bethesda, MD 21814-5044

Dr. Michael Letsky
Office of the Chief of Naval
Operations (OP-01B7)
Washington, DC 20350

Lt. Dennis McBride
Human Factors Branch
Pacific Missile Test Center
Point Mugu, CA 93042

Dr. George Moeller
Human Factors Department
Naval Submarine Medical Res Lab
Naval Submarine Base
Groton, CT 06340-5900

Capt. W. Moroney, USN
Naval Air Development Center
Code 602
Warminster, PA 18974

Naval Aerospace Medical
Research Laboratory
Sensory Division, Code 23
Pensacola, FL 32508

Dr. A.F. Norcio
Computer Sciences & Systems
Code 5592
Naval Research Laboratory
Washington, DC 20375-5000

Office of the Chief of Naval Operations
OP-933D3
Washington, DC 20350-2000

Office of Naval Research
Perceptual Science Program (3 copies)
Code 1142 PS
800 North Quincy Street
Arlington, VA 22217-5000

Dr. W.A. Rizzo
Head, Human Factors Division
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3224

Dr. Randall P. Shumaker
NRL A.I. Center
Co[illegible]10
U.[illegible] Center for ARAI
V[illegible] 20375-5000

LC[illegible]. Singer
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. James G. Smith
Office of Naval Research
Code 1121
800 N. Quincy Street
Arlington, VA 22217-5000

Special Assistant for Marine
Corps Matters
Code OOMC
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

DEPARTMENT OF THE ARMY

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Michael Kaplan
Director, Office Basic Res
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Milton S. Katz
U.S. Army Research Institute
ARIESCO
Box 65
FPD, NY 09510-1500

Dr. John Weisz
Technical Director
U.S. Army Human Engineering Laboratory
Aberdeen Proving Ground, MD 21005

DEPARTMENT OF THE AIR FORCE

Dr. Charles Bates, Director
Director, Engineering Division
USAF AAMRL/HES
Wright-Patterson AFB, OH 45433

Dr. Kenneth R. Boff
AF AMRL/HE
Wright-Patterson AFB, OH 45433

Dr. Alfred R. Fregly
AF Office of Sci. Res.
Life Sciences Directorate
Bldg. 410
Bolling AFB, DC 20332-6448

Dr. John Tangney
AF Office of Sci. Res.
Life Sciences Directorate
Bldg. 410
Bolling AFB, DC 20322-6448

OTHER GOVERNMENT AGENCIES

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (2 copies)

Dr. Richard Loutitt
National Science Foundation
Division of Behavioral & Neural Sciences
1800 G. Street NW
Washington, DC 20550

OTHER ORGANIZATIONS

Dr. H.E. Bamford, Program Director
Division of Information,
Robotics and Intelligent Systems
National Science Foundation
1800 G Street, N.W.
Washington, DC 20550

Prof. Richard Catrambone
School of Psychology
Georgia Tech
Atlanta, GA 30332

Dr. Marvin S. Cohen
Dec. Sci. Consortium, Inc.
1895 Preston White Drive
Suite 300
Reston, VA 22091

Dr. Thomas Duffy
Audio-Visual Center
Indiana University
Bloomington, IN 47405

Prof. James H. Howard, Jr.
Department of Psychology
Human Performance Lab
Catholic University
Washington, DC 20064

Dr. Bonnie E. John
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. William B. Johnson
Search Technology, Inc.
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092

Prof. David L. Kleinman
Electrical Engineering & Computer Science
Department
University of Connecticut
Storrs, CT 006268

Dr. William H. Muto, Manager
Corporate Human Factors, R&D
Texas Instruments, Inc.
MS 8223
P. O. Box 655474
Dallas, TX 75265

Dr. Allen Newell
Department of Computer Science
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
Experimental Psych. Dept.
70 Fawcett Street
Cambridge, MA 02238

Dr. Scott Robertson
Department of Psychology
Rutgers University
Busch Campus
New Brunswick, NJ 08903

Dr. William B. Rouse
Search Technology, Inc.
4725 Peachtree Corners Circle #200
Norcross, GA 30092

Professor Penelope M. Sanderson
1206 W. Green Street
Department of Mechanical and
Industrial Engineering
Urbana, IL 61801

Dr. H. Wallace Sinaiko
Manpower Research Advisory Services
Smithsonian Institution
801 N. Pitt Street, Suite 120
Alexandria, VA 22314-1713

Ms. Piyawadee Sukaviriya
Department of Electrical
Engineering and Computer Science
The George Washington University
Washington, DC 20052

Dr. Douglas Towne
U. of Southern California
Behavioral Technology Lab
1845 South Elena Avenue
Fourth Floor
Redondo Beach, CA 90277

Prof. William R. Uttal
Department of Psychology
Arizona State University
Tempe, AZ 85287-1104

Dr. H.P. Van Cott
NAS-National Research Council
(COHF)
2101 Constitution Avenue, NW
Washington, DC 20418

Prof. Christopher D. Wickens
Department of Psychology
University of Illinois
Urbana, IL 61801